



Windfall

第29回世界コンピュータ将棋選手権アピール文章

作成：井本 康宏 作成日：2019/3/吉日

忙しい人のためのチートシート

これさえあれば、1分で全てわかる

▶ 今回のWindfallは以下の要素で成り立っています

• 仕様：

- 合法手からランダムに次の手を選ぶ
- (実装が間に合えば、評価値が最良のものを選ぶ)

• プログラミング：

- TensorflowからPytorchに変更
- 局面の遷移、合法手の計算をYaneuraOuを参考に実装

• その他：

- なぜか去年よりさらに後退した進捗
- 一緒に開発してくれる方を募集中

← 去年も書いた

わからなかった方は、次項以降をどうぞ

開発者自己紹介

▶ 開発者

- ▶ 井本 康宏 (twitter: @Windfall_shogi)

▶ 職業

- ▶ エンジニア

▶ 棋力

- ▶ 測ったことがないのでわかりませんが、すごく弱いです
- ▶ 学生時代に囲碁・将棋部だったなんて言えない

▶ 開発のきっかけ

- ▶ 趣味、好奇心、研究スキル、プログラミングスキルの向上を目的として、当時、話題だったこともあり開発を始めました

去年より-3歩前進

- ▶ DNN関係で大幅な見直し
- ▶ この1年での変遷
 - ▶ 状態行動価値関数から状態価値関数に変更
 - ▶ DQNは行動の種類が多過ぎて無理
 - ▶ 大駒の利きを考慮するためには畳み込み層が多く必要なので、全結合層で遠くの駒も考慮して計算
 - ▶ 次元の呪いやばい
 - ▶ 長い利きの方向ごとに畳み込みを計算(例えば1x9のカーネルで畳み込み)
 - ▶ このタイミングでPytorchに変更
 - ▶ gatherやscatterがいい感じ

長い利きの方向ごとの畳み込み

- ▶ 広い範囲の畳み込みを行うことで、レイヤーが少なくても学習できる
 - ▶ 十字飛車を考えるとかなり広い範囲の計算が必要なので、3x3のカーネルでは辛そう
- ▶ 例えば、縦方向なら1x9のカーネルでストライドは0なので、実質的には全結合
- ▶ ラインごとに別のカーネルで計算
 - ▶ ラインとは、例えば、縦方向なら一つの筋のこと
 - ▶ 全て同じカーネルでは制約が強すぎると思う
 - ▶ gatherで集める→畳み込み(1x1 → 1x9 → 1x1)→scatterで9x9に並べなおす
 - ▶ メモリのコピーがかなりボトルネック
 - ▶ gather, scatterと同時に1x1の畳み込みを実行
 - ▶ pytorchのextensionでカスタムレイヤーを自作
 - ▶ ボトルネックは少し改善したが、各ラインの処理が逐次的に実行されるのであまり速くない

評価関数について

- ▶ Residual Networkを採用
 - ▶ 持ち駒を入力データのチャンネルではなく、畳み込み後のバイアス項に対して計算
- ▶ 浅いネットワークから徐々に深くして学習
 - ▶ 同時に出力も大雑把なものから精緻なものへ段階的に学習
 - ▶ 長い利きを考慮した畳み込みレイヤーを使って学習
 - ▶ 浅いネットワークでも長い利きを学習する機会があるのはメリットだと思う
- ▶ 現状は挙動が明らかにおかしいので、間に合わないかも
 - ▶ 本番用PCで動かないのはつらい
 - ▶ ユニットテストは通るが、DNNに組み込むとエラー

探索について

- ▶ 今年もそこまで手が回らなかった
 - ▶ 今年は、合法手の中からランダムか評価値が最も高いものを選ぶ
- ▶ 本来の実装したい手法の詳細は最低限のものが出来上がった時に書きます
 - ▶ 過去のアピール文書で論文を参照しているので、どうしても気になる方は探してください

大会後の方針

- ▶ 長い利きの畳み込みレイヤーが動くなら
 - ▶ 本来の探索アルゴリズムの実装
 - ▶ DNNを深くしながら逐次的に学習
- ▶ 動かないなら
 - ▶ GPUは面倒が多いので、NNUEを改良しようかな
 - ▶ policyベースの学習とか面白そう
 - ▶ NNUEの本質は3駒以上の特徴量を手に入れたことで、非線形性ではないと思う
 - ▶ 具体的な方針は未定